ManiplIcons in ThinkerToy

Steven Hugh Gutfreund
Computer and Information Science Department
University of Massachusetts

COINS Technical Report 87-30

ManiplIcons in ThinkerToy

Steven Hugh Gutfreund*
GTE Laboratories
40 Sylvan Road
Waltham, Massachusetts 02254
(617)466-2933 sgutfreund@gte.com (csnet)

July 17, 1989

Abstract

ThinkerToy is a graphical environment for modeling decision support problems. It provides a tableau on which such problems as landscape planning, service scheduling, and statistical analysis can be modeled and analyzed. Normally, complex mathematical and statistical modeling techniques are needed to perform meaningful analysis. ThinkerToy uses graphical icons with concrete physical properties to replace mathematical relationships and properties. The key construct in this methodology is the **ManiplIcon**: an icon which is not just a pictorial representation, but also a semantic tool for building models which homorphically represent semi-structured problems.

*This work was supported by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, and the Air Force Office of Scientific Research, Bolling AFB, DC 20332 under Contract No. F30602-85-C-0008. This contract supports the Northeast Artificial Intelligence Consortium (NAIC). Additional support was provided by the Tektronix Artificial Intelligence Marketing group in the form of a generous equipment grant.

1 Definition

What is ThinkerToy? Very briefly, ThinkerToy can be defined as a *concrete modeling* environment for construction of *reasoning tableaus* for *decision support* problems. However, in order to fully understand this capsule definition we need to elucidate each italicized term.

	Management Activity			
Type of Decisional Task	**Operational Control**	**Management Control**	**Strategic Planning**	**Support Needed**
Structured	Inventory Reordering	Linear Programming for Manufacturing	Plant Location	Clerical Systems
Semistructured	Bond Trading	Setting Budgets for Advertising	Capital Acquisition Anal.	DSS
Unstructured	Selecting a Cover for TIME magazine	Hiring a Manager	R & D Budgeting	Human Intuition

Table 1: A Framework for Decision Support Systems. Courtesy of: Addison-Wesley Publishing Company [44, page 87]

1.1 Decision Support

Decision problems can either be thoroughly structured, unstructured, or a combination of the two (Table 1) [44]. Well structured problems such as inventory reordering, equipment scheduling, cash flow analysis, or meeting scheduling, are recurring problems that are largely clerical in nature. For these problems there exist many computer algorithms and programs. For unstructured problems such as hiring personnel or designing a corporate logo, computational support is impossible because the problems involve subjective intuitive evaluations. For semistructured problems such as financial planning, upgrading manufacturing facilities, or architectural design, intuition is insufficient, yet there are no comprehensive solutions by computer programs. Programs may assist, but comprehensive solutions are beyond simple computational techniques.

Semistructured problems require a *decision support system* (DSS). This system should provide support for both elements of semistructured problems: structured and unstructured. It should provide an environment for the decision maker to interactively analyze and explore a problem in order to obtain insights. Also, it should supply quantitative data to validate any conclusions drawn. The best environments for such tasks are modeling environments. With a model a decision maker can obtain insights into the workings of his problem and, in addition, the model can be used to produce quantitative results. However, the modeling technique that is chosen must be appropriate for both the representation and the interactive analysis of semistructured problems.

1.2 Symbolic Modeling

There are two basic methods for modeling decision support problems: symbolic models and concrete models. A symbolic modeling technique, such as linear programming, uses symbolic or mathematical equations to represent a system. Analysis of symbolic models is performed by mathematical procedures such as Simplex, sensitivity analysis, and dynamic programming. However, symbolic models have difficulties in both the representation and analysis of semistructured problems.

First, it is necessary to force the model to fit within the bounds of a tool that is intended for handling structured problems. Usually this means that one must translate the model into mathematical formulae. Expressing the model in terms of arithmetic equations may cause incorrect or inappropriate quantification of parts of the model. Auditorium design and the siting of nuclear power plants, for instance, are two types of problems where major parts of the puzzle cannot be encoded in terms of mathematical formulae.

Second, when the model is in a quantified form it is difficult to explore design alternatives. It is usually difficult enough to keep track of these alternatives when one's mental representation of the problem and the model of the problem are the same. However, when the model is encoded symbolically one must also worry about stray symbolic and computational effects which are extraneous to the selection of a design alternative. In addition, each design alternative must be translated into the symbolic model and the results translated back into the decision maker's mental representation. This double translation has been one of the major obstacles in the the use of analytic decision methods.

Finally, symbolic models are simply inappropriate for solving problems with large visual components. A traffic planner exploring traffic flow, a police superintendent planning police beats, an architect designing physical plant floor plans, are all examples of semistructured problems that have large visual components. It would be self-defeating to force such problems into non-visual mathematical representations.

1.3 Concrete Modeling

Concrete modeling uses a graphic to represent a system. For example, a map can be used to represent routing and spatial distribution problems. Concrete modeling is not the duplication of a physical system in terms of a graphical diagram. In order to be useful, a concrete model must extract the relevant features of a system into a concise graphical depiction. For example, the best map is not one that is identical with the model. If one is lost in a forest, a map that showed you exactly what you saw would be of little use. Instead, the best map is one that employs schematic representations yet preserves most of the spatial components of the original. Therefore, the key to producing quality concrete models is the use of concise graphic schematics.

Analysis of a concrete model involves spatial inference. The level of sophistication of the spatial inferencing that is possible in a concrete model depends on the richness of the graphic representation in that model. Spreadsheets, Simplex tables, and Karnaugh maps are grid based graphic representations. They only permit spatial inferencing based on the placement and layout of symbolic data. Line charts, histograms, and relational charts are two-dimensional graphic representations. On these concrete models statisticians and stockbrokers can employ more sophisticated techniques such as shape fitting and shape recognition. Terrain maps and thematic maps are multidimensional graphic representations. On them one can analyze traffic flow, measure environmental impact, and perform cost and risk assessment of large scale construction projects (e.g. dams, turnpikes, airports, etc.).

Concrete modeling is a very good technique for semistructured problems. For problems which are inherently graphical and spatial there is a very close match between the users original mental model and the concrete model. Even for non-visual abstract problems, if one can convert the problem to a visual representation, the solution to the problem is easier to perceive.

Admittedly, symbolic models have had advantages over concrete models — they have been easier to computerize and consequently it has been easier to extract quantitative data from them. As a result, they are much more common. However, recent technological advances have removed the major implementation difficulties involved in concrete models. The availability of computerized concrete models will ease the extraction of quantitative data and lessen the general bias against the use of concrete models.

1.4 Reasoning Tableaus

A reasoning tableau is a graphical representation of a system (a concrete model) coupled with a set of tools and operations. By means of these tools and operations a decision maker can visually manipulate the information contained in the concrete model in a manner that assists the formation of insights into DSS problems. Examples of paper and pencil reasoning tableaus include: matrixes, Simplex tableaus, Pert charts, spreadsheets, and Karnaugh maps. Usually these tableaus are used only for displaying data. However, clever people have discovered that they can also be used for exploratory data analysis in semistructured problems. Graphic heuristics can be used in Karnaugh maps to minimize gates, Simplex tableaus can be employed to explore the sensitivity of a transportation network to shipping costs fluctuations, and spreadsheets can be tinkered with to investigate investment strategies.

However, paper and pencil reasoning tableaus are very primitive. It is a very time-consuming task to draw a graphic reasoning tableau. It is an even more tedious task to redraw the tableau after the application of a spatial manipulation or analysis tool. Therefore most reasoning tableaus are limited to grid based representations of information and vector operations.[1] ThinkerToy, on the other hand, is an environment for constructing general purpose reasoning tableaus. ThinkerToy can be used to create sophisticated multidimensional concrete models that employ rich graphical representations of concrete and abstract information. To this has been coupled a wide variety of sophisticated spatial manipulation tools. For example in ThinkerToy it is possible to create decisions support environments for:

STOCKBROKERS: Stock market charts that a stockbroker can use curve fitting tools to interactively fit head and shoulders curves.[2]

STATISTICIANS: Statistical charts which statisticians can use for factor analysis.

[1] Or, in the case of more sophisticated concrete models such as terrain and thematic maps, the graphical manipulation tools have have been restricted so that redrawing will not have to be done.

[2] Technical analysts look for a three hump shape in stock price charts that resembles a head with a shoulder on each side. By fitting this shape to a stock price chart technical analysts can discover stock price trends that indicate price breakouts [73,83]. The fitting of a shape is a task that involves human intuition and insight, since there is no rule for the time span nor the nature of the shape involved. Therefore, a fixed mathematical test cannot be developed. However, in ThinkerToy one can create a visual template and apply it to a range of stocks.

CIVIL ENGINEERS: Models of coastal plains and river deltas which can be flooded to determine the effects of hurricanes and other storms.

ACOUSTIC ENGINEERS: A model that simulates an orchestra hall and has tools for changing seating layout, wall placement, and cooling system ducts — coupled with decibel metering tools for determining the acoustic effect of any changes.

With spreadsheets one can perform simple arithmetic operations on columns and rows of data. ThinkerToy, on the other hand, provides much greater power by allowing any graphic manipulation, computation, or selection over any kind of visual representation of information. Within the ThinkerToy environment it is easy to reproduce the functionality of simpler tableaus such as Simplex, Karnaugh maps, spreadsheets, RESQ [59], Helix® [31], and Stella® [34].

1.5 ThinkerToy

To return to the original definition: *ThinkerToy is a concrete modeling environment for construction of reasoning tableaus for decision support problems.* It is a decision support environment because it is intended for exploring semistructured problems. It is a concrete modeling environment because it employs concrete modeling techniques to depict decision support problems. And it is a reasoning tableau because exploration and analysis of decision support problems is done by employing graphical tools and operations that are visually and semantically coupled to a graphical representation of the system.

These three descriptions: decision support environment, concrete modeling environment, and an environment for the construction of reasoning tableaus constitute a capsule definition of ThinkerToy. Within this definition there exist many subsidiary concepts. In order to grasp the scope and value of the ThinkerToy approach it is necessary to identify these subsidiary concepts. For this more comprehensive study, I refer the reader to my thesis [29]. In this paper I will concentrate on the description of one aspect of the ThinkerToy system, the ManiplIcon.

2 ManiplIcons in ThinkerToy

ThinkerToy is a homogeneous object oriented system where every object is a graphical entity. But this homogeneity goes beyond merely creating a picture world. Objects not only look concrete, they also "feel" concrete. Every object is directly manipulatable [81,82]. Together these objects create a language. This is a language whose grammatical rules are formed by the tactile feel of its constituents, and whose semantics and syntax are revealed by the visual metaphors it employs. Thus, the very actions and verbs of the language are couched in visual manipulative metaphors. This homogeneity extends from the very lowest to the very highest parts of the system:

0d Scalars: Tools for Trig functions, Log functions, detection, injection, and applying values.

1d Arrays: Tools for ripping out, injecting, and overlaying values on the face of tabular data.

2d Charts: Tools for shape fitting, axis stretching, and extraction nets.

3d TerrainMaps: Tools for physical and pseudo-physical molding and growing of features on terrain and thematic maps.

The key construct for accomplishing this homogeneity are Panels (also referred to as ManiplIcons). A ManiplIcon is an active icon whose actions are invoked via manipulating it with a mouse. There are Panels that represent a broad set of the basic Smalltalk object family: Integer, Float, Symbol, Array, Form. There are also composite Panels that represent tools useful in building tableaus: Scales, Charts, Thermostats, Buttons, ControlBoards, ControlPanels, Toolkits, etc.

ManiplIcons carry out their actions via direct manipulation (via the mouse). I decompose direct manipulation into two properties:

1. Visual Analysis Procedures (VAPs). One can apply a ManiplIcon to another to carry out analysis, e.g. A ruler over a map, A Least-Squares curve fitter over a Chart of data points, an OperationPanel (LOG base 10) over a NumArray.

2. Scripts. VAPs correspond to the application of a tool, and a script to the characteristics of how one "picks up and handles" the tool. Scripts are a language for mouse interaction. Activities such as dragging, picking, lassoing, stretching, are handled in a generic manner via scripts.

Another way to explain the ManiplIcon metaphor is by the Toolbox analogy. In a toolbox one has hammers, screwdrivers, pliers, saws, bevels, grouters, paintbrushes, etc. In order to use a tool one must pick it up and adjust it, then apply it to an object. The pick-up and adjust operation corresponds to a script. An example of this is the picking up and adjustment of a vise-grip wrench. The visual application phase would correspond to the actual act of beveling, painting, or loosening.

3 Examples of ManiplIcons in ThinkerToy

Four environment domains have been built in ThinkerToy. Perhaps none of them are extensive enough to form a comprehensive tool for true professional decision makers. However, they do form a kernel of and environment which a team of domain experts at an OEM[3] could build a comprehensive support environment. The four graphic tableaus are:

Array: A tableau for storing a group of panels. They can be NumberPanels, TextPanels, Charts, Maps, etc. This tableau was created largely as a tutorial. Both to introduce a user to ThinkerToy and to explore a straight forward implementation of a spreadsheet using the ManiplIcon metaphor.

Chart: A tableau for the analysis of 2-Dimensional data. This domain served as the beginning of an exploration of the use of spatial analysis to explore data structure. There are spatial tiling tools, regression analysis filters, shape fitting tools for least squares analysis, and semantically active nets for extracting data by attribute.

TerrainMap: A tableau for the analysis of 3-Dimensional data. This domain attempts to show the advantages of spatial analysis techniques. There are tools for distance (rubber rulers for as-the-crow-flies, as-the-crow-walks, as-the-tired-hiker-walks), radiation (visible features), growth (crystal growth based map contents), differentiation (slope, velocity gradient), and stream (steepest downhill path).

[3]It seems that in an attempt to hide past sins OEMs (Original Equipment Manufacturers) have started to call themselves VARs (Value Added Retailers).

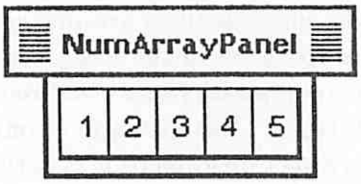

Figure 1: NumArray

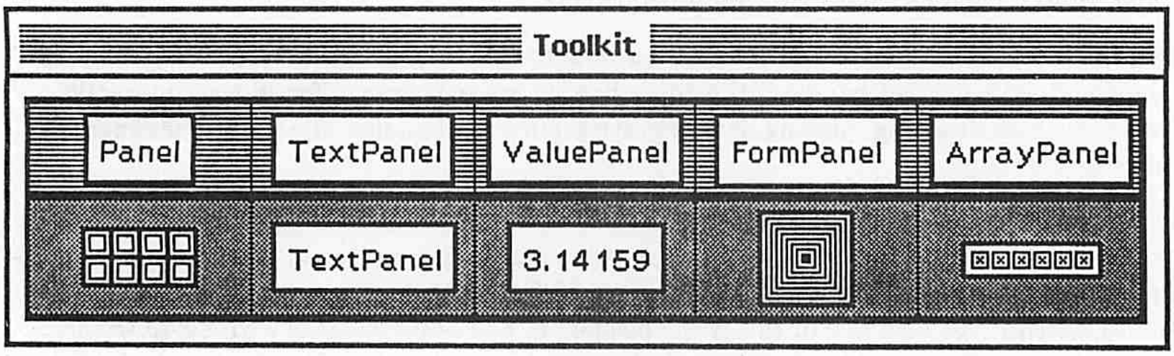

Figure 2: Toolkit

DataFlow: A tableau for flow rate problems and modeling ThinkerToy code graphically. Telephone network queuing, chemical rate functions, data-flow in satellite signal processing, data-flow in a computer, or data-flow in an a packet delivery system. This tableau is related to commercial products such as: Stella® [34], XA-1000 [79], and BTL's Performance Analysis Workstation [62].

3.1 ArrayPanels

The simplest way to use an ArrayPanel is to store things such as numbers (fig. 1) or tools (fig. 2).[4] To manipulate a panel one must mouse it and ask for its ControlPanel (fig. 3).

Before we go further with this description, it is necessary to introduce some terminology and fill in some background. Every object in the ThinkerToy system is a subclass of Panel. Panel implements the basic functionality of a ManiplIcon. New panels are created by using an old panel as a template and adding new subPanels. To manipulate a panel we bring up a ControlPanel. The ControlPanel has three parts: a tableau (the object being manipulated), a standardControlBoard (a set of standard operation buttons below the tableau) and a mainControlBoard (object specific operation buttons next to the tableau).[5]

[4] Actually the Toolkit has two arrays, a bottom row of bins and a top row of binNames.

[5] It is only a minor implementation limitation that prevents having more that two ControlBoards. Ideally, one has a ControlBoard for each category of messages, and can shuffle through them.

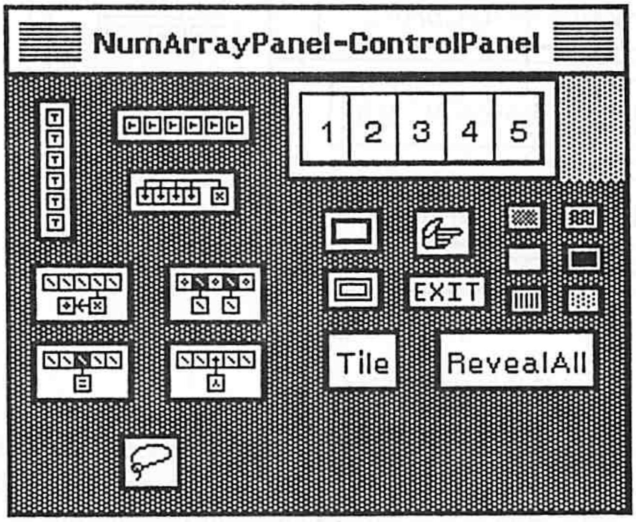

Figure 3: ControlPanel for NumArray

A Button, (fig. 4) describes the VAP and the script. There are two ways to describe the operation of a Button. Using the toolbox metaphor, each script describes a tactile interaction scheme for picking up and handling the tool. The VAP describes the intrinsic operation that the tableau is to perform. When a button is pressed the script is invoked to pick up and adjust an accessory tool that will then be used to perform the VAP.

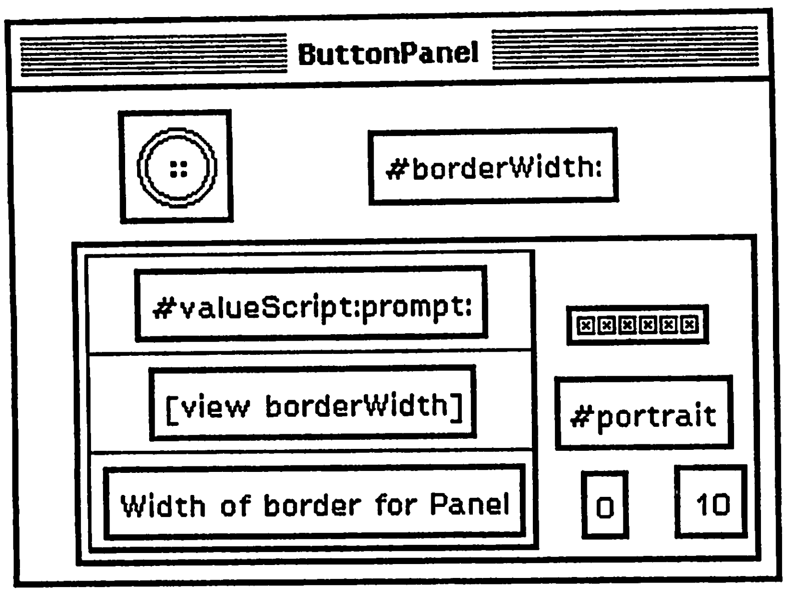

Figure 4: ButtonPanel

At the semantic level, viewing ThinkerToy as a graphical object oriented language, the VAP is the method selector and the script is the means for selecting the arguments. Just as in normal languages actual arguments will vary from call to call, so do script interactions allow one to indicate and vary the passing of arguments.[6]

We can now return to the discussion of ArrayPanels. The ArrayPanel domain was intended to serve as a tutorial for both me and the user of the ThinkerToy system. It provides a container for a sequence of objects and a tableau for applying a tool over this sequence. Since the Smalltalk methods: **collect:**, **inject:into:**, **detect:**, and **select:** create strong visual images of tools operating over a collection, I decided to implement these as the VAPs' of a ArrayPanel (fig. 5). Figure 6 shows a few of the kinds of accessory tools one can apply over an ArrayPanel. Continuing in this manner several ArrayPanels can be cascaded together to produce a spreadsheet. The advantages to this approach to spreadsheet making are:

- One can fill in the cells with any sort of nonhomogenous objects (pictures, arrays, lists, buttons, ControlPanels, etc.).

[6]These two levels of metaphor are only indications of deeper and more pervasive structure and layering within ThinkerToy. Another parallel metaphor is represented by the ControlBoard. Each ControlBoard acts as category of methods of the Smalltalk browser.

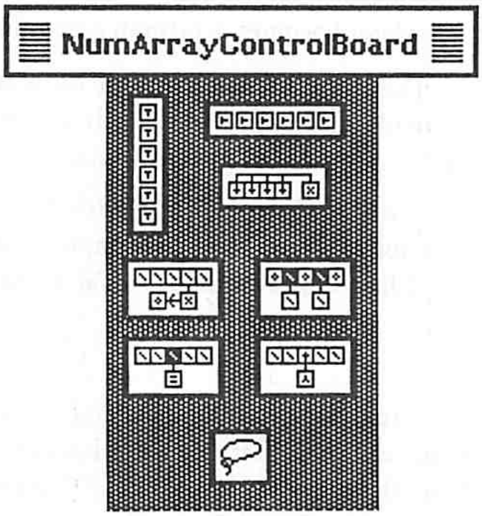

Figure 5: ControlBoard for NumArray

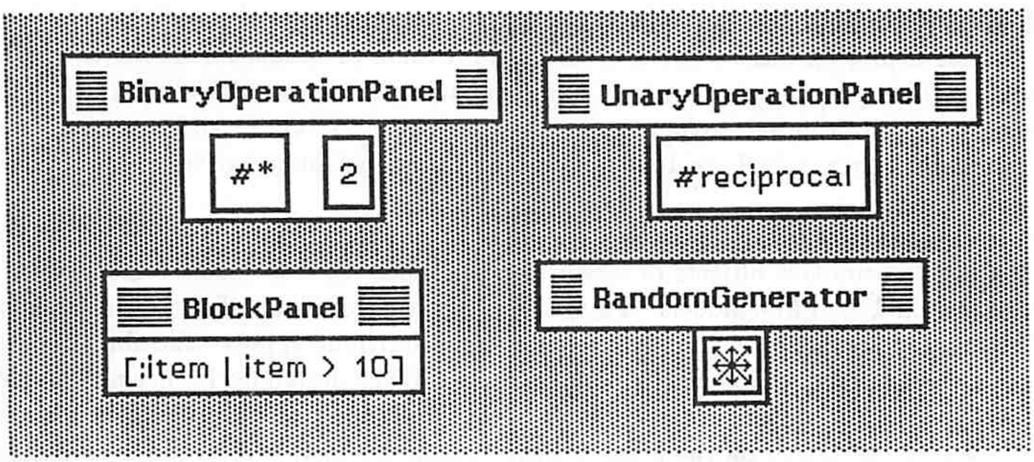

Figure 6: Tools that operate on Arrays

- Spreadsheets are basically limited to a few fixed functions. In this scheme newly innovated tools can be applied to the spreadsheet as they are thought up. As a consequence of this, if the cells are made to represent pixels and the tools are the MacPaint® paintCans and sprayCans then the ArrayPanel tableau becomes a bitmap editor like MacPaint®.
- Spreadsheets are still limited in their ability to deal with data in a physical manner. A ManipIIconic spreadsheet can provide lassos, content sensitive overlays, and other objects which can take advantage of spatial structure of gridded data.

However, I tend to view spreadsheets as a fairly mundane environment for attempting to create concrete models and graphical manipulations. Therefore in this implementation I limit ArrayPanels to one dimensional operations, and use Charts for two dimensional modeling.

3.2 Charts

A Chart is a tableau for displaying and analyzing two dimensional data. The markers on a chart (data points) can be any object with any number of attributes. However, when they are placed on the chart two attributes are selected for the x and y coordinates. Consequently, charts are a kind of graphic relational database that have VAPs for joining and projecting. Charts can be used in Cartesian or polar modes and thus they can be applied to simultaneously show a star chart and elliptical orbits of the planets. However, the main illustration of chart power comes from exploring graphic VAPs for statistical analysis.

Multi-variate statistical analysis involves looking for trends, coherences, and structure in data. Recently, researchers such as John Tukey [35,92] and Edward Tufte [89,90,91] have been actively involved in advancing the graphic tools of statisticians. Constructs such as box-plots (fig. 7) [16] can graphically reveal the structure of data.

My chief aim has been to try and turn these constructs for graphical annotation into a kind of physical tool for exploring data structure. In figure 8 there are tools for fitting shapes (least squares, lowess) grabbing percentile chunks of data, spreading content sensitive nets (e.g. pull out all class M stars), and filtering out structure (e.g. regression analysis). In addition, the axes of a chart are composed of ArrayPanels so that one is able to apply OperationPanels such as Log to produce log charts.

This last example brings out two important points. First, the components of ThinkerToy are meant to serve as the primitive objects of larger constructions. Charts themselves are later used as metering devices in DataFlow models. ThinkerToy is a kind of TinkerToy for professionals to explore and discover. Second, since ThinkerToy is a tool for decision support - the ability to explore and discover facts in an extemporaneous manner is crucial. When Johannes Kepler discovered that $T^3 = R^2$ he did it by pouring through the tables of Tycho Brahe's observations. In ThinkerToy one would grab a star chart, pull off the planets, project them by their orbital radius and period of revolution on another chart, then cube root one axis and square root the other to observe the data structure. Further data structure could be revealed by regression analysis.

3.3 TerrainMaps

In civil engineering, landscape architecture, and regional government problems arise that involve logistic planning, water resources, land use planning, traffic flow, etc. All these problems are multidimensional problems with strong spatial and graphic components.

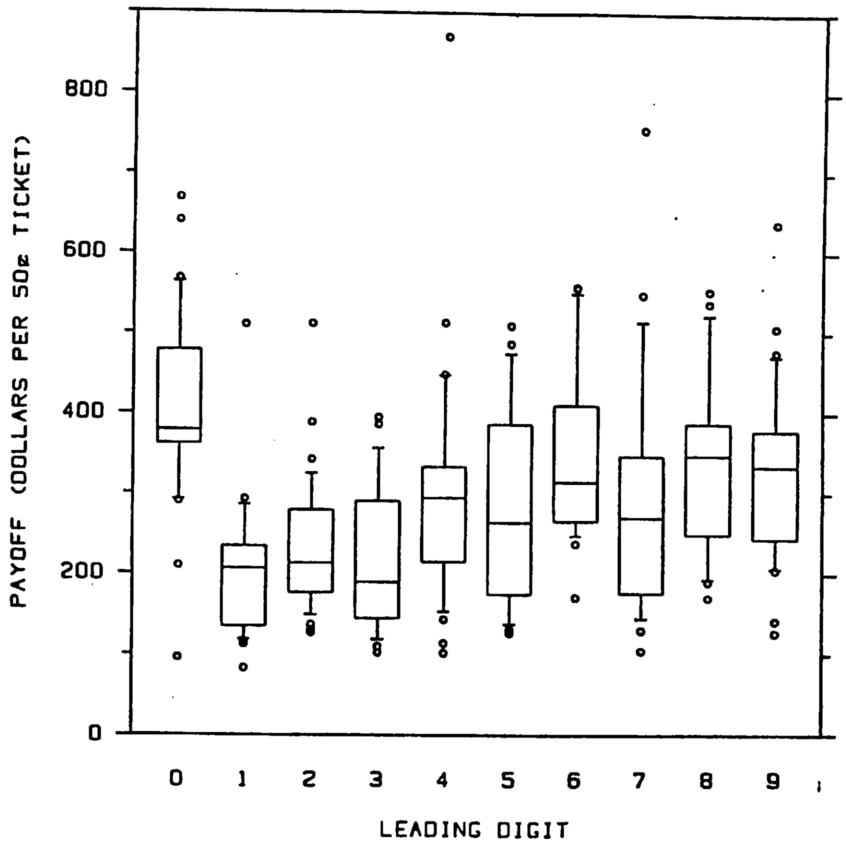

Figure 7: Tukey Whisker Box (box plot)

TerrainMaps (fig. 9) provide a reasoning tableau for modeling terrain (height) pseudo-terrain (cost-height, velocity-height, noise-height) and thematic features (water, soil composition, forest, roads). While many different kinds of tools can be created to explore problems in this domain, I chose to implement those that inspired images of spatial manipulation and molding (fig. 10). Some, such as spread distance, rubber rulers, differentiating height and velocity slopes, and draining to find optimal path density are inspired by MAP [8,88]. Others, such as crystals that grow context sensitive features and graftals were inspired from other sources [85]. Some arose spontaneously from my exploration of terrain models. This was the case with Vehicles. I began to realize that it would be nice to have some sort of programmatic control of crystal growth. With a set of scripts one loads the vehicle with some feature paint (e.g. road), attaches headlights (convolution sensors for heading, slope, feature density), and installs a driver (to weight the sensors and establish an end condition). The result is a tool for exploring road construction based on maintaining bearing, elevation, or cost.

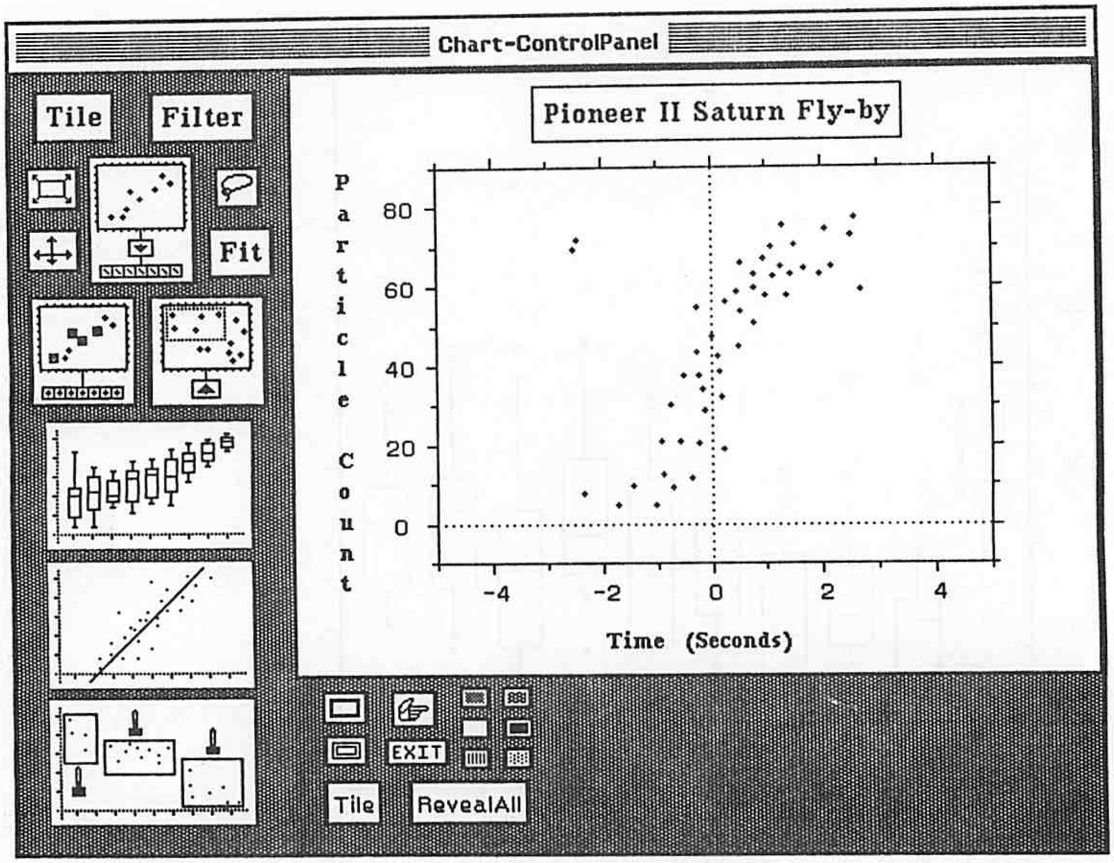

Figure 8: Chart

3.4 DataFlow

ThinkerToy is an environment where all components of the system are accessible and use the same graphic metaphor. Thus, the code that implements the various VAPs also has a representation in terms of graphic tableaus. In addition, the same tableau can be used to represent flow models. Examples of flow models are: traffic light queues, file servers, chemical rate functions, packet delivery systems, and supermarket queues.

In a flow model (fig. 11) each CommandButton (the disk with two black wedges) takes a method selector and its arguments, sends it to an object and returns the result. CommandButtons can either have explicit numerical sequence or operate on data availability (data flow). The result is a kind of graphical object oriented language.[7] In this example we have three deli counters and four checkout counters acting as servers. Each one has a service queue protected by monitors. In figure

[7]Though it is very simple, and while many improvements could be made, there is some very interesting power in this model. One can see the entire flow of execution and can interrupt, explore and modify (via ControlPanels) any level of the code, just like the Smalltalk inspector and debugger. Furthermore, even though this is a level of the system intended for expert craftsmen who build new environments and not decision makers themselves, even at this level one is dealing with the same graphic language and not some implementation language such as Lisp or Smalltalk.

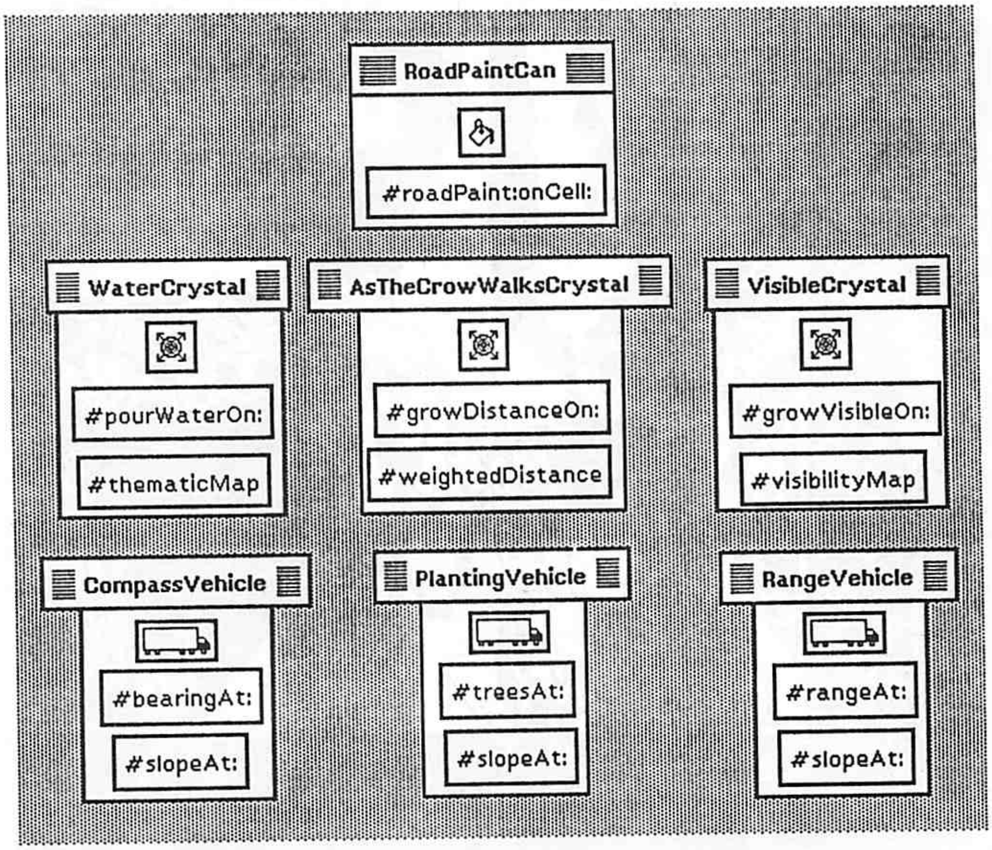

Figure 10: Tools for operations on TerrainMaps

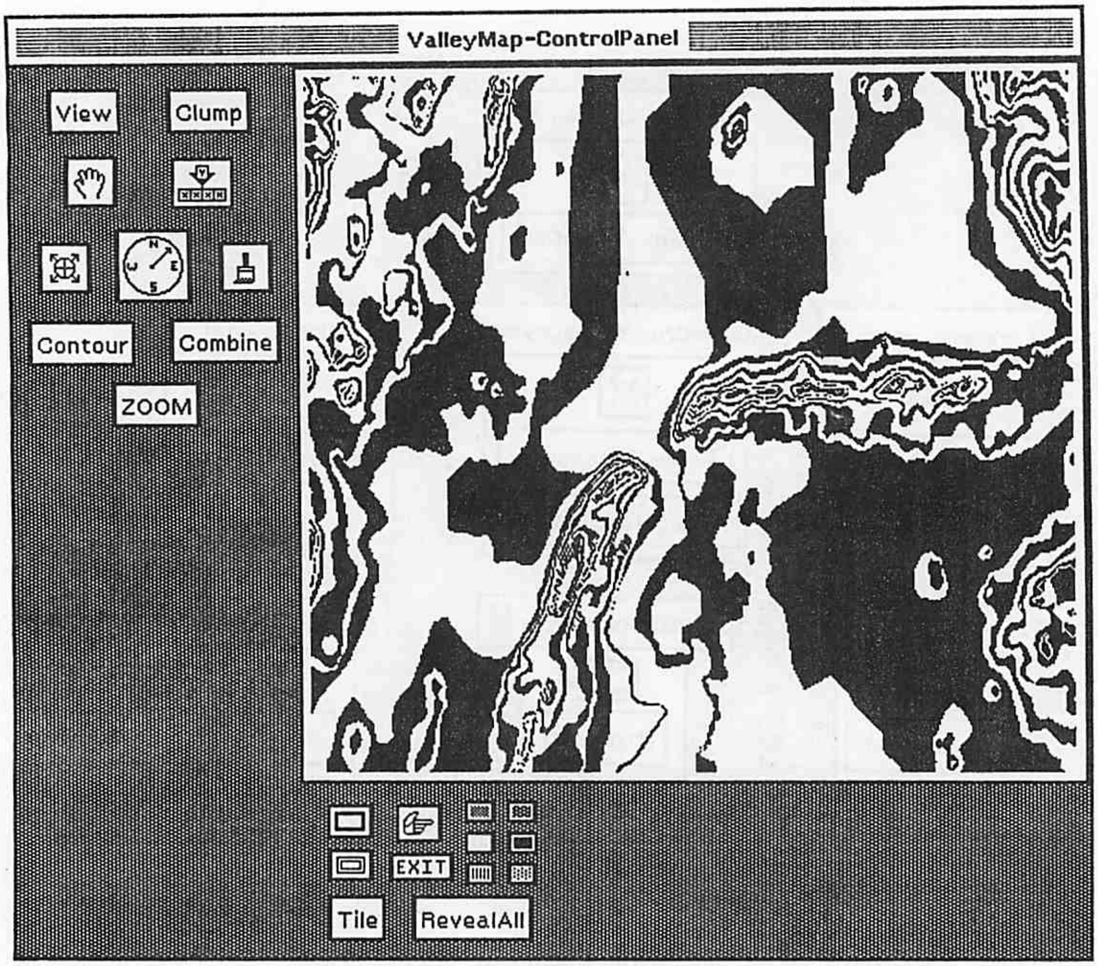

Figure 9: ControlPanel for TerrainMap of Pioneer Valley, MA

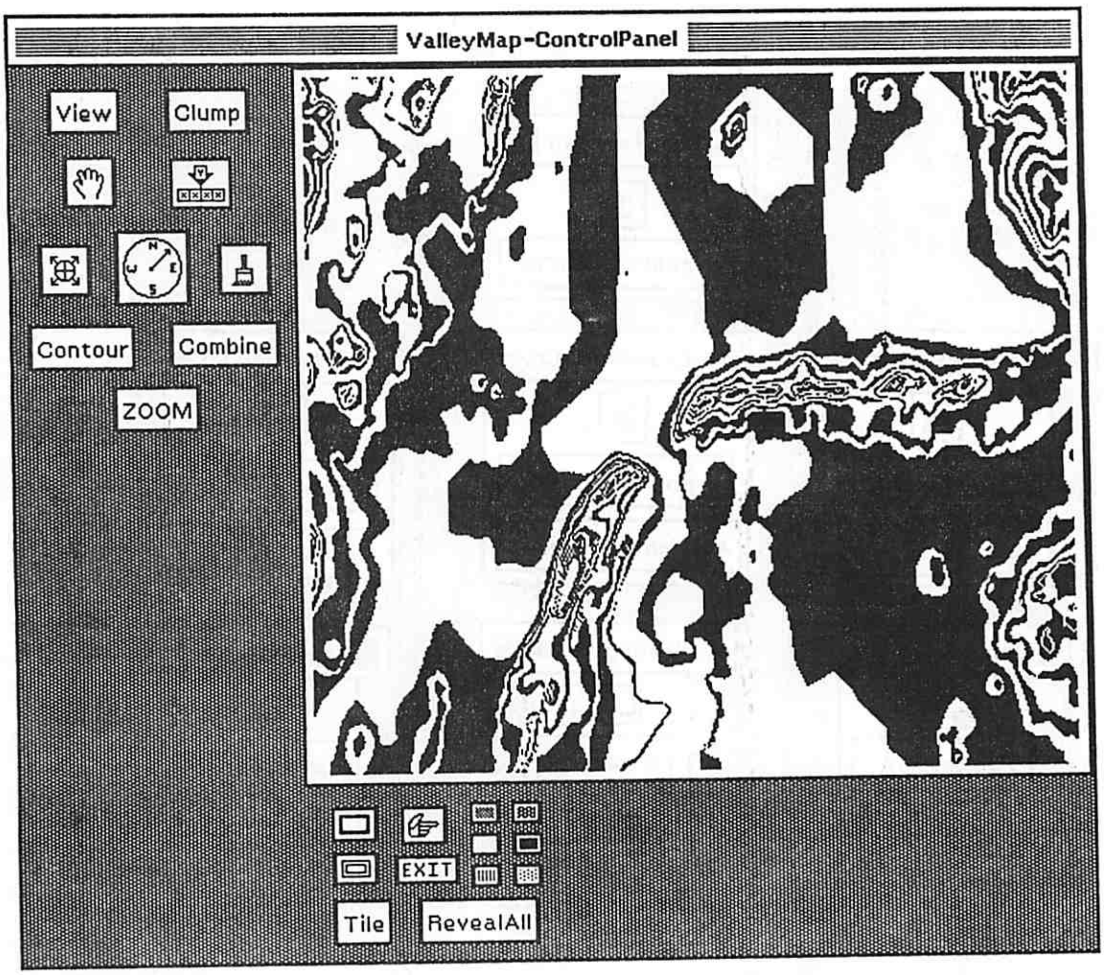

Figure 9: ControlPanel for TerrainMap of Pioneer Valley, MA

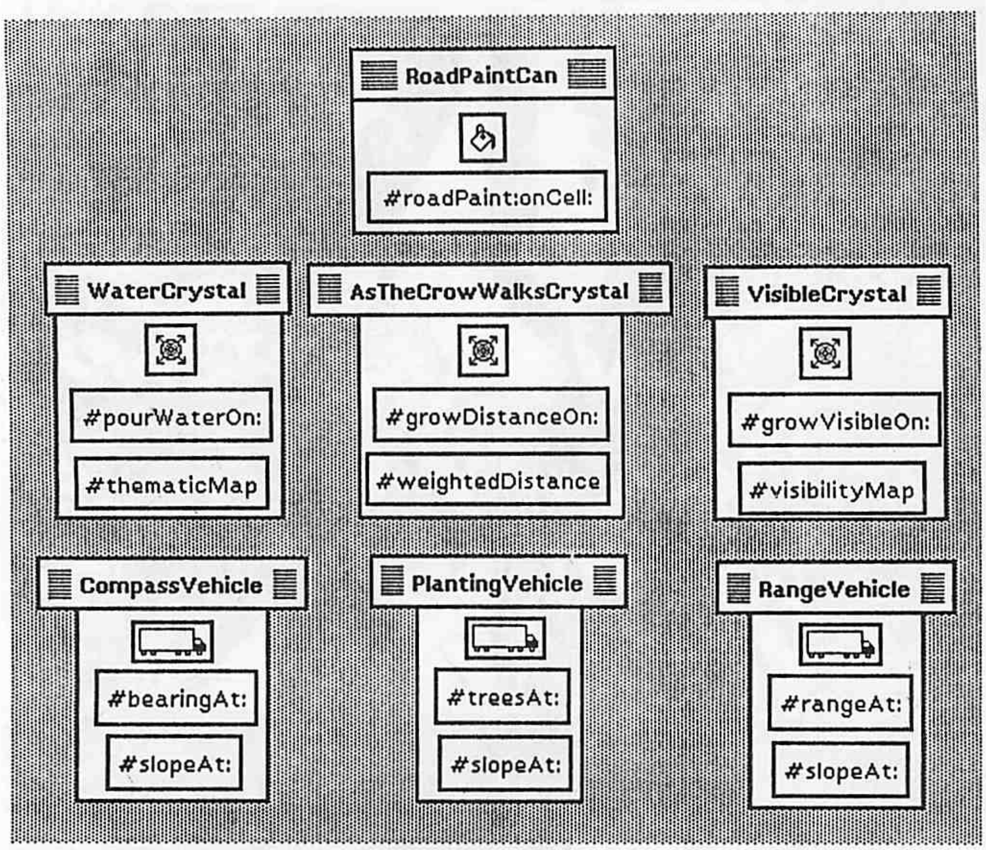

Figure 10: Tools for operations on TerrainMaps

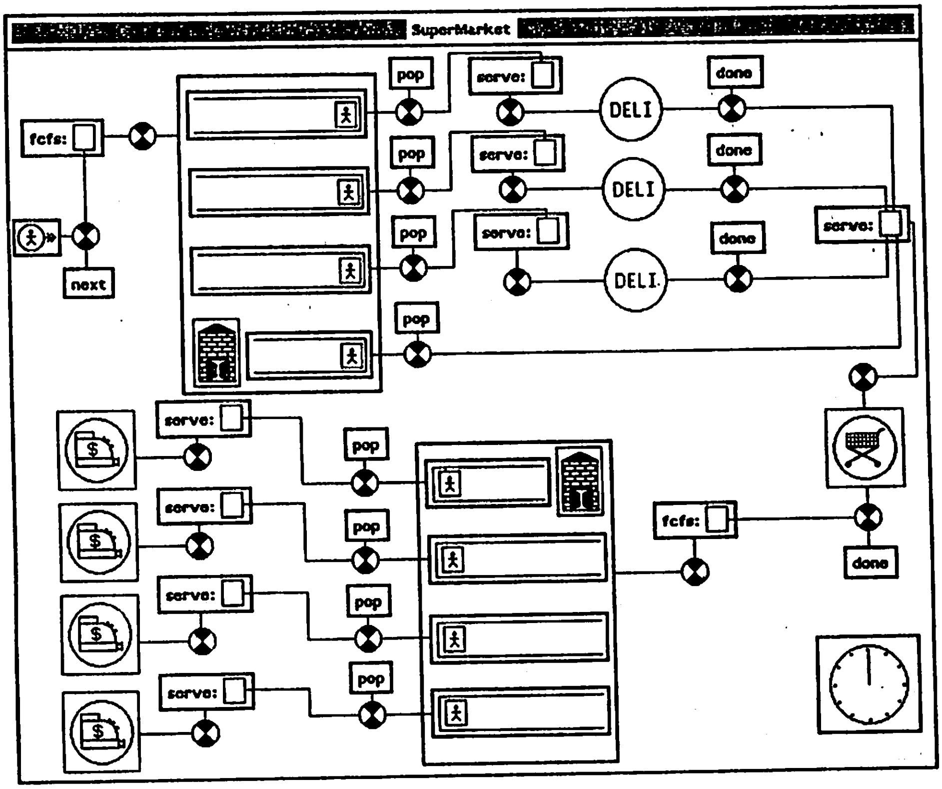

Figure 11: SuperMarket with express checkout

12 there is a barber shop where the servers are protected by the monitors and we are using a chart as a strip chart data recorder.

4 Conclusions & Analysis

The ThinkerToy implementation consists of four tableaus. This kernel is meant to form the foundation for constructing large systems. Interesting applications begin to occur when one creates a FlowModel, instruments it with charts as meters, collects data from experiments, uses the chart to perform statistical analysis, and then performs iterative changes via arrays. Only when one begins to use all components together and then produces hybrid models from this base, does the power of the ThinkerToy environment become apparent.

There remain many areas for further research. However, it is useful to also look backwards and identify areas for improvement.

One problem seems to be the use of buttons for invoking operations. This seems to limit the spontaneity of the interface and probably is not much of an improvement over pull down menus. While it served well as an implementation compromise, the original ThinkerToy plan called for a rigorous characterization of the grasping and manipulations that one can simulate with a mouse. These actions (grasp, drag, stretch, rub) would be used to directly indicate the manipulation being performed.

On the positive side, ThinkerToy has the ability to capture the functionality of the rapidly growing market of graphic tableaus. Within the ThinkerToy environment one can simulate the interface and function of Odesta Helix® [31], MacPaint®, spreadsheets, RESQ [59], ARK [87], MAP [8,88], Stella® [34], PAW [62], and XA-1000 [79]. While the interface may not be as carefully tuned as these fixed function tools, it is certainly comforting to find that one could replicate the functionality of a tool such as RESQ in one week.

5 Acknowledgments

I wish to thank my thesis committee: Prof. David McDonald, Prof. Bruce Croft, and Prof. Francis Hill for their criticism and support. I also want to express my appreciation to Prof. Victor Lesser for financial assistance and Chuck Smith at Tektronix for the tools that made this work possible.

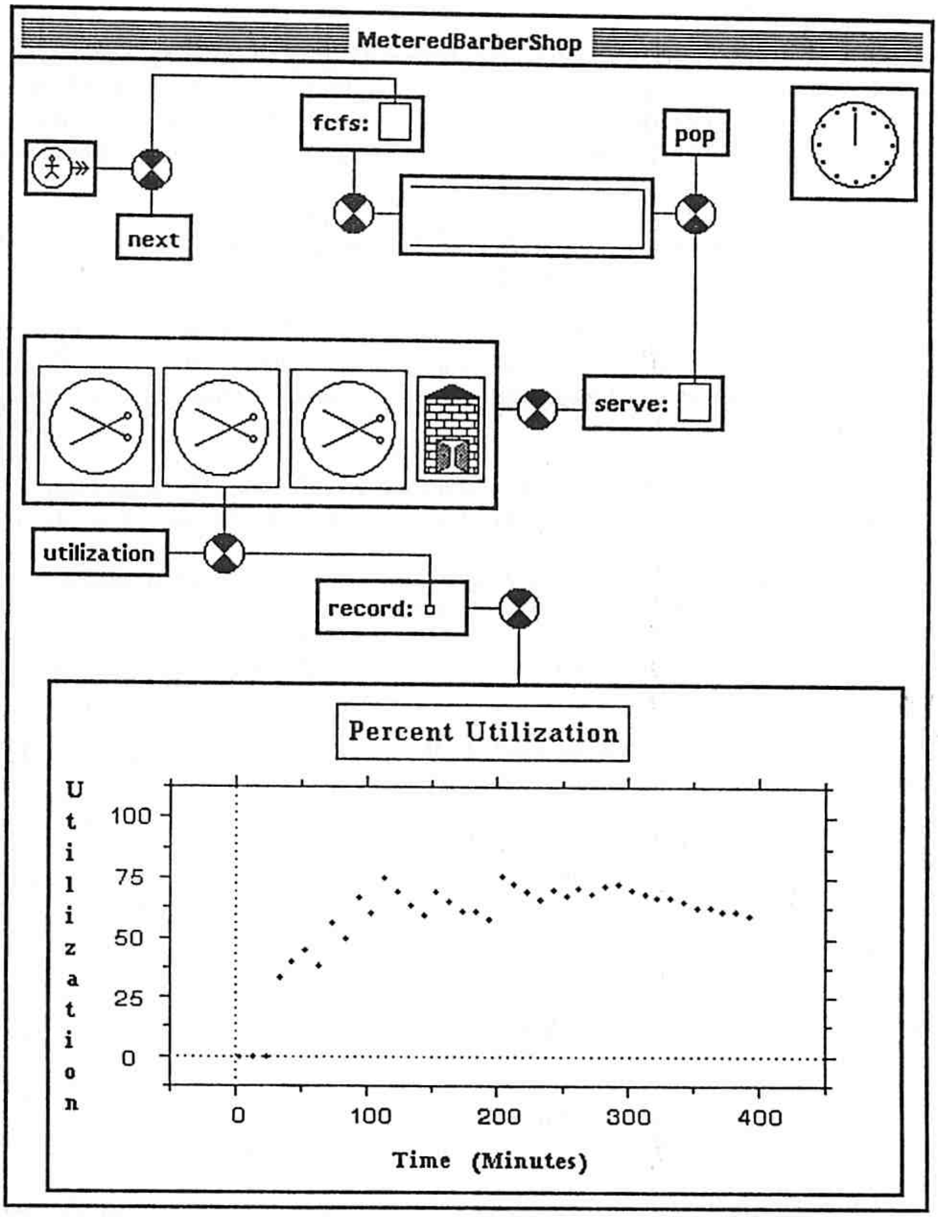

Figure 12: BarberShop with metering

References

[1] Steven Alter, "Decision Support Systems: Current Practice and Continuing Challenges", Addison-Wesley, Reading, Massachusetts, 1980.

[2] Guiseppe Attardi and Maria Simi, *Extending the Power of Programming by Examples*, appearing in: "Integrated Interactive Computing Systems", edited by: Pierpaolo DeGano and Erik Sandewall, Proceedings of the European Conference on Integrated Interactive Computing Systems, ECICS 82, Stresa, Italy, 1 – 3 September, 1982, North-Holland Publishing Co., Amsterdam, Netherlands, 1983, pp. 3 – 25.

[3] Roberto Bachi, "Graphical Rational Patterns: A New Approach to Graphical Presentation of Statistics", Israel University Press, Jerusalem, Israel, 1968.

[4] Roberto Bachi, *Proposals for the Development of Selected Graphical Methods*, appearing in: Graphical Presentation of Statistical Information, Bureau of the Census, Technical Paper No. 43, 136 Annual Meeting of the American Statistical Association, Social Statistics Section, Boston, Massachusetts, August 23 – 26, 1976, pp. 23 – 68.

[5] Ron Baecker, *Sorting out Sorting*, 16mm color, sound film, 25 minutes, Dynamics Graphics Project, Computer Systems Research Institute, University of Toronto, Toronto, Ontario, Canada, 1981.

[6] David R. Barstow, Howard E. Shrobe, and Erik Sandewall, "Interactive Programming Environments", McGraw-Hill Book Company, New York, New York, 1984.

[7] Jacques Bertin, "Graphics and Graphic Information Processing", Walter de Gruyter, Berlin, Germany, 1981.

[8] Joseph K. Berry, "Geographic Information Analysis Workshop Workbook", School of Forestry and Environmental Studies, Yale University, New Haven, Connecticut, 1984.

[9] K. S. Bhaskar, J. K. Peckol, and J. L. Beug, *Virtual Instruments: Object Oriented Program Synthesis*, OOPSLA'86 Object Oriented Programming Systems, Languages and Applications, September 29 – October 2, 1986, Portland, Oregon, Norman Meyrowitz editor, **SIGPLAN Notices**, Vol. 21, No. 11, (November 1986), pp. 303 – 314.

[10] Eric Allan Bier and Maureen C. Stone, *Snap-Dragging*, SIGGRAPH'86 Conference Proceedings, Edited by David C. Evans and Russell J. Athay, August 18 – 22, 1986, Dallas Texas, **Computer Graphics**, Vol. 20, No. 4, (August 1986), pp. 233 – 240.

[11] Alan Borning, *ThingLab — A Constraint Oriented Simulation Laboratory*, Technical Report No. SSL-79-3, Xerox Palo Alto Reserach Center, 3333 Coyote Hill Road, Palo Alto, California, (July 1979).

[12] Alan Borning, *Defining Constraints Graphically*, in Proceedings of CHI'86 Human Factors in Computing Systems, Boston, Massachusetts, April 13 – 17, pp. 137 – 143.

[13] Marc H. Brown and Robert Sedgewick, *A System for Algorithm Animation*, **Computer Graphics**, Vol. 18, No. 3, (July 1984), Proceedings of SIGGRAPH'84, Minneapolis, Minnesota, pp. 177 – 186.

[14] Gretchen P. Brown, Richart T. Carling, Christopher F. Herot, David A. Kramlich, and Paul Souza, *Program Visualization: Graphical Support for Software Development*, **IEEE Computer**, Vol. 18, No. 8, (August 1985), pp. 27 – 35.

[15] Luca Cardelli, *Two-Dimensional Syntax for Functional Languages*, appearing in: *Integrated Interactive Computing Systems*, Proceedings of the European Conference on Integrated Interactive Computing Systems, ECICS 83, Stresa, Italy, 1-3 September, 1982 North-Holland Publishing Co., Amsterdam, Netherlands, 1983. pp. 139 – 151.

[16] William S. Cleveland, "The Elements of Graphing Data", Wadsworth Advance Books and Software, Monterey, California, 1985.

[17] John J. Coyle and Edward J. Bardi, "The Management of Business Logistics", West Publishing Co., St. Paul, Minnesota, 1980.

[18] S. Peter DeJong and Roy J. Byrd, *Intelligent Form Creation in the System for Business Automation (SBA)*, IBM Research Report, No. RC 8579, IBM T. J. Watson Research Center, Yorktown Heights, New York, (10/24/80).

[19] Andrea A. DiSessa and Harold Abelson, *Boxer: A Reconstructable Computation Medium*, **Comunications of the ACM**, Vol. 29, No. 9, (September 1986), pp. 859 – 870.

[20] Robert A. Duisberg, *Animated Graphical Interfaces*, in Proceedings of CHI'86 Human Factors in Computing Systems, Boston, Massachusetts, April 13 – 17, pp. 131 – 136.

[21] Harold M. Elliot, *Cardinal Place Geometry*, **Geographical Analysis**, Vol. 17, No. 1, (January 1985), Ohio State University Press, pp. 16 – 35.

[22] Craig Field and Nichlos Negroponte, *Using New Clues to Find Data*, Proceedings of the Third International Conference on Very Large Data Bases, Tokyo, (1977), pp. 156 – 158.

[23] Mark Friedell, "Automatic Graphics Environment Synthesis", Ph.D. Thesis, Department of Computer Engineering and Science, Case Western Reserve University, Technical Report No. CCA-83-03, Computer Corporation of America, Four Cambridge Center, Cambridge, Massachusetts 02142, 1983.

[24] Ephraim P. Glinert and Steven L. Tanimoto, *Pict: An Interactive Graphical Programming Environment*, **IEEE Computer**, Vol. 7, No. 11, (November 1984), pp. 7 – 28.

[25] Adele Goldberg and David Robson, "Smalltalk-80: The Language and its Implementation", Addison-Wesley, Reading, Massachusetts, 1983.

[26] Earnst H. Gombrich, "Art and Illusion; a Study in the Psychology of Pictorial Representation", Princeton University Press, Princeton, New Jersey, 1961.

[27] Laura Gould and William Finzer, *Programming by Rehersal*, Technical Report No. SCL-84-1, Xerox Palo Alto Research Center, (May 1984).

[28] Steven H. Gutfreund, *Environments for Decision Support*, Technical Report No. 84-10, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts, 1984.

[29] Steven H. Gutfreund, "ThinkerToy: An Environment for Decision Support", Ph.D. Thesis, Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003, 1988.

[30] Frederick Hartwig and Brian E. Dearing, "Exploratory Data Analysis", Sage Publications, Beverly Hills, California, 1979.

[31] Jim Heid, *New Relations* Macworld, Vol. 3, No. 5, (May 1986), pp. 100 – 107.

[32] Walter Herdeg, "Graphis Diagrams: The Graphic Visualization of Abstract Data", Graphis Press Corporation, Zürich, Switzerland, 1983.

[33] Christoper F. Herot, Richart Carling, Mark Friedell, and David Kramlich, *A Prototype Spatial Data Management System*, **Computer Graphics**, Vol. 14, No. 3, (July 1980), SIGGRAPH'80 Conference Proceedings, pp. 63 – 70.

[34] High Performace Systems, *Stella*, software program, graphic modeling software for the MacIntosh, High Performace Systems, 13 Dartmouth College Highway, Lyme, New Hampshire 03768

[35] David C. Hoaglin, Frederick Mosteller, and John W. Tukey, "Exploring Data Tables, Trends, and Shapes", John Wiley & Sons, New York, 1985.

[36] Kristina Spargren Hooper, *The Use of Computer-Controlled Videodisks in the Study of Spatial Learning*, **Behavior Research, Method and Instrumentation**, Vol. 13, No. 2, (1981), pp. 77 – 84.

[37] Lawrence J. Hubert, Reginald C. Golledge, Michael Costanzo, and Nathan Gale, *Measuring Association Between Spatially Defined Variables: an Alternative Procedure*, **Graphical Analysis**, Vol. 17, No. 1, (January 1985), Ohio State University Press, pp. 36 – 46.

[38] Robert J. K. Jacob, *A State Transition Diagram Langauge for Visual Programming*, **IEEE Computer**, Vol. 18, No. 8, (August 1985), pp. 51 – 59.

[39] Spencer A. Joyner, "A Method for Integrating Selected Landscape Values into Sytems Planning", Ph.D. Thesis, Department of Civil Engineering, University of Massachusetts, Amherst, Massachusetts, 1978.

[40] Alan Kay, "The Reactive Engine" Ph.D. Thesis, University of Utah, Salt Lake City, Utah, September, 1969.

[41] Alan C. Kay, *Microelectronics and the Personal Computer*, **Scientific American**, Vol. 237, No. 3, (September 1977), pp. 230 – 244.

[42] Alan C. Kay and Adelle Goldberg, *Personal Dynamic Media*, **IEEE Computer**, Vol. 10, No. 3, (March 1977), pp. 31 – 41.

[43] Alan Kay, *Computer Software*, **Scientific American**, Vol. 251, No. 3, (September 1984), pp. 53 – 59.

[44] Peter G. W. Keen and Michael S. Scott Morton, "Decision Support Systems: An Organizational Perspective", Addison-Wesley, Reading, Massachusetts, 1978.

[45] Peter G. Keen, *DSS: An Executive Mind-Support System*, **Datamation**, Vol. 25, No. 12, (November 1979), pp. 117 – 126.

[46] Peter G. W. Keen, *Decision Support Systems: A Research Perspective*, CISR Technical Report No. 54 and Sloan W.P. No. 1117-80, Center for Information Systems Research, Sloan School of Business, Massachusetts Institute of Technology, Cambridge, Massachusetts, (March 1980).

[47] Peter G. W. Keen and Thomas J. Gambino, *Building a Decision Support System: The Mythical Man-Month Revisited*, CISR Technical Report No. 57 and Sloan W.P. No. 1132-80, Center for Information Systems Research, Sloan School of Business, Massachusetts Institute of Technology, Cambridge, Massachusetts, (May 1980).

[48] Peter G. W. Keen, *Decision Support Systems and Managerial Productivity Analysis*, CISR Technical Report No. 60 and Sloan W.P. No. 1156-80, Center for Information Systems Research, Sloan School of Business, Massachusetts Institute of Technology, Cambridge, Massachusetts, (October 1980). pp. 117 - 122.

[49] Philip Klahr, John W. Ellis, Jr., William Giaria, Sanjai Narain, Edison M. Cesar, Jr., and Scott R. Turner, *TWIRL: Tactical Warfare in the ROSS Language*, Report No. R-3158-AF, Rand Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, California 90406, (October 1984).

[50] Arthur Koestler, "The Art of Creation", Dell Publishing Co., New York, 1964.

[51] Stephen Michael Kosslyn, "Image and Mind", MIT Press, Cambridge, Massachusetts, 1980.

[52] Alex Kramer, *Icon Maker: Interactive User Interface Design*, IEEE Computer Society, 1984 Workshop on Visual Languages, December 6 - 8, 1984, Hiroshima, Japan, pp. 192 - 198.

[53] A. S. Levens, "Graphical Methods in Research", John Wiley & Sons, New York, 1965.

[54] Henry Lieberman and Carl Hewitt, *A Session with TINKER: Interleaving Program Testing with Program Design*, MIT AI Memo, No. 577, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, (September 1980).

[55] Henry Lieberman, *Seeing What Your Programs Are Doing*, MIT AI Memo, No. 656 , Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, (February 1982).

[56] Henry Lieberman, *There's More to Menu Systems Than Meets the Screen*, **Computer Graphics**, Vol. 19, No. 3, (July 1985), Proceedings of SIGGRAPH'85, San Francisco, California, pp. 181 - 189.

[57] Ralph L. London and Robert A. Duisberg, *Animating Programs Using Smalltalk*, **IEEE Computer**, Vol. 18 No. 8, (Aug 1985), pp. 61 - 71.

[58] Kevin Lynch, "The Image of the City", MIT Press, Cambridge, Massachusetts, 1960.

[59] Edward A. MacNair and Charles H. Sauer, "Elements of Practical Performance Modeling", Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[60] David McArthur, Philip Klahr, and Sanjai Narain, *ROSS: An Object-Oriented Language for Constructing Simulations*, Report No. R-3160-AF, Rand Corporation, 1700 Main Street, P.O. Box 2138, Santa Monica, California 90406, (December 1984).

[61] David M. McKeown, Jr. and Jerry L. Denlinger, *Graphical Tools for Interactive Image Interpretation*, **Computer Graphics**, Vol. 16, No. 3, (July 1982), SIGGRAPH'82 Conference Proceedings, pp. 189 - 198.

[62] Benjamin Melamed and Robert J. T. Morris, *Visual Simulation: The Performance Analysis Workstation*, **IEEE Computer**, Vol. 18 No. 8, (Aug 1985), pp. 87 - 94.

[63] Arthur I. Miller, "Imagery in Scientific Thought: Creating 20th Century Physics", Birkhäuser, Boston, Massachusetts, 1984.

[64] Margaret Minsky, *Manipulating Simulated Objects with Real-World Gestures Using a Force and Position Sensitive Screen*, **Computer Graphics**, Vol. 18, No. 3, (July, 1984), Proceedings of SIGGRAPH'84, Minneapolis, Minnesota, pp. 195 - 203.

[65] Mark Morconi and Dwight F. Hare, *Visualizing Program Design Through Pegasys*, **IEEE Computer**, Vol. 18 No. 8, (Aug 1985), pp. 72 - 85.

[66] Brad A. Myers, *Visual Programming, Programming by Example, and Program Visualization: A Taxonomy*, in Proceedings of CHI'86 Human Factors in Computing Systems, Boston, Massachusetts, April 13 - 17, pp. 59 - 66.

[67] William M. Newman and Timothy Mott, *OFFICETALK-Zero: An Experimental Integrated Office System*, appearing in: *Integrated Interactive Computing Systems*, Proceedings of the European Conference on Integrated Interactive Computing Systems, ECICS 83, Stresa, Italy, 1-3 September, 1983 North-Holland Publishing Co., Amsterdam, Netherlands, 1983. pp. 315 - 331.

[68] Seymour Papert, "Mindstorms: Children, Computers and Powerful Ideas", Basic Books, New York, New York, 1980.

[69] Kurt W. Piersol, *Object Oriented Spreadsheets: The Analytic Spreadsheet Package*, OOPSLA'86 Object Oriented Programming Systems, Languages and Applications, September 29 - October 2, 1986, Portland, Oregon, Norman Meyrowitz editor, **SIGPLAN Notices**, Vol. 21, No. 11, (November 1986), pp. 385 - 390.

[70] John H. Pittman and Donald P. Greenberg, *An Interactive Graphic Environment for Architecture Energy Simulation*, Proceedings of SIGGRAPH'82, **Computer Graphics**, Vol. 16, No. 3, (July 1982), pp 233 - 234.

[71] Man-Chi Pong, *2-PIGS: An Interactive Graphical Programming Environment with Mixed Interpretation and Compilation*, Laboratory Report No. 22, UKC Computing Laboratory, University of Kent, Kent CT2 7NF, Canterbury 66822, (September 1984).

[72] Man-Chi Pong, *I-PIGS: A Concurrent Programming Environment with Interactive Graphical Support*, Laboratory Report No. 26, UKC Computing Laboratory, University of Kent, Kent CT2 7NF, Canterbury 66822, (November 1984).

[73] Martin J. Pring, "Techinical Analysis Explained", McGraw Hill, New York, 1980.

[74] Georg Raeder, *A Survey of Current Graphical Programming Techniques*, **IEEE Computer**, Vol. 18 No. 8, (Aug 1985), pp. 11 - 25.

[75] Steven P. Reiss, *Graphical Program Development with PECAN Program Development Systems*, **SIGPLAN Notices**, Vol. 19, No. 5, (May 1984), Proceedings of the ACM SIGSOFT/SIGPLAN Software Engineering Symposium on Practical Software Development Environments, pp. 30 - 41.

[76] Derek Rogers, "Creative Systems Design", Anbar Publications LTD., Wembley, England, 1970.

[77] Larry H. Rowe, *Programming Language Issues for the 80's*, **SIGPLAN Notices**, Vol. 19, No. 8 (August 1984), pp. 51 - 61.

[78] Hannan Samet, *The Quadtree and Related Hierarchical Data Structures*, **Computer Surveys**, Vol. 16, No. 2, (June 1984), pp. 187 - 260.

[79] Martin H. Schrage and Damian F McArdle, *New Array Processor Continuous Simulation System Uses Tactile Sensing Icon Programming*, 1986 Proceedings Summer Computer Simulation Conference, Society for Computer Simulation, pp. 138 - 142.

[80] W. Schild, L. R. Powers, and M. Karnaugh, *Picture World: A Concept for Future Office Systems*, IBM Research Report, No. RC 8384, IBM T. J. Watson Research Center, Yorktown Heights, New York, (7/30/80).

[81] Ben Schneiderman, *Direct Manipulation: A Step Beyond Programming Languages*, **IEEE Computer**, Vol. 16, No. 8, (August 1983), pp. 57 – 69.

[82] Ben Schneiderman, *The Future of Interactive Systems and the Emergence of Direct Manipulation*, appearing in "Human Factors and Interactive Computer Systems", Proceedings of the NYU Symposium on User Interfaces, New York, May 26 – 28, 1984, Yannis Vassiliou editor, Ablex Publishing Corporation, Norwood, New Jersey, 1984.

[83] William F. Sharpe, "Investments", 2nd edition, Prentice-Hall, New York, 1981.

[84] Herbert A. Simon, "The New Science of Management Decision", 3rd edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1977.

[85] Alvy Ray Smith, *Plants, Fractals, and Formal Languages*, **Computer Graphics**, Vol. 18, No. 3, (July, 1984), Proceedings of SIGGRAPH'84, Minneapolis, Minnesota, pp. 1 – 10.

[86] David Canfield Smith, "Pygmalion: A Computer Program to Model and Stimulate Creative Thought", Birkhäuser Verlag, Basel und Stuttgart, 1977.

[87] Randall B. Smith, *The Alternative Reality Kit: An Animated Environment for Creating Interactive Simulations*, 1986 IEEE Computer Society Workshop on Visual Languages, June 25 – 27, 1986, Dallas Texas, pp. 99 – 106.

[88] C. Dana Tomlin, "Digital Cartographic Modeling Techniques in Environmental Planning", Ph.D. Thesis, School of Forestry and Environmental Studies, Yale University, New Haven, Connecticut, 1984.

[89] Edward R. Tufte, "The Quantitative Analysis of Social Problems", Addison-Wesley Publishing Co., Reading, Massachusetts, 1970.

[90] Edward R. Tufte, "Data Analysis for Politics and Policy", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1974.

[91] Edward R. Tufte, "The Visual Display of Quantitative Information", Graphic Press, Chesire, Connecticut, 1983.

[92] John W. Tukey, "Exploratory Data Analysis", Addison-Wesley Publishing Co., Reading, Massachusetts, 1977.

[93] Michael Williams, James Hollan, and Albert Stevens, *An Overview of STEAMER: An Advanced Computer-Assisted Instruction System for Propulsion Engineering*, **Behavior Research, Method and Instrumentation**, Vol. 13, No. 2, (1981), pp. 85 – 90.

[94] Edward A. Williams and Alison K. Massa, "Siting of Major Facilities: A Practical Approach", McGraw Hill Inc., New York, 1983.

[95] Moshe. M. Zloof, *Query-by-Example*, AFIPS Conference Proceedings, National Computer Conference, Vol. 44, (1975), pp. 431 – 438.

[96] Moshe. M. Zloof and S. P. deJong, *The System for Business Automation (SBA)*, **Communications of the ACM**, Vol. 20, No. 6, (June 1977), pp. 385 – 396.

[97] Moshe M. Zloof, *Office-by-example: A business language that unites data and word processing and electronic mail*, **IBM Systems Journal**, Vol. 21, No. 3, (1982), pp. 272 – 304.